

CLAIMS:

1. A method of generating data for a three-dimensional (3D) virtual environment using a map of 3D components, the method comprising:

reading map data from a data store, the map data comprising component identifying data and component position data for at least one of said 3D components; reading component data for the at least one identified component from a data store, the component data including at least 3D geometry data for the component; and transforming the 3D geometry data of the at least one component using said component position data to provide 3D virtual environment data for said 3D virtual environment, said 3D virtual environment data defining a substantially contiguous 3D surface enclosing said 3D virtual environment.

2. A method as claimed in claim 1 wherein said map data comprises component identifying data and component position data for a plurality of said components, the method further comprising reading and transforming 3D geometry data for said plurality of components and combining said transformed data for said plurality of components to provide said 3D virtual environment data.

3. A method as claimed in claim 2, wherein each of said components has at least one interface portion for interfacing to the interface portion of another component, and wherein said component data includes two versions of said 3D geometry data for at least the interface portion of the component, a non-interfaced version of the geometry data and an interfaced version of the geometry data, and wherein the method further comprises:

reading the non-interfaced version of the geometry data for the interface portion of a component where the component is joined to another component at the interface portion.

4. A method as claimed in claim 3, wherein the 3D components comprise a set of 3D components at least a first subset of which share substantially matching interfaced versions of interface portion geometry.

09852126-050901

5. A method as claimed in claim 4, wherein the set of 3D components comprises a second subset of components with matching interface geometry data, at least one component having two interface portions with first and second interfaced versions of geometry matching the interface geometry of the respective first and second subsets of components.

6. A method as claimed in claim 1, wherein said component data for a component identified in the map data includes plug data for at least one plug for closing an interface portion of the component; the method further comprising transforming the plug data using component position data for the component, to close the interface when said interface of the component is not used.

7. A method as claimed in claim 2, wherein the 3D geometry data of said plurality of components comprises data for generating visible surface portions of said contiguous 3D surface enclosing said 3D virtual environment.

8. A method as claimed in claim 2, wherein the three-dimensional virtual environment comprises an environment for a computer game character, wherein the 3D geometry data comprises data for generating visible 3D geometry of the 3D virtual environment, and wherein the component data includes additional game operation-related data for defining an operational aspect of the game; the method further comprising:

combining the game operation-related data for components identified in the map data to operationally link parts of the 3D virtual environment derived from different components.

9. A method as claimed in claim 8, wherein said game operation-related data includes position data, and wherein the method further comprises transforming said game operation-related data using said component position data before combining the game operation-related data.

10. A method as claimed in claim 8, wherein said game operation-related data includes collision geometry data, and wherein a result of said combining includes data

09852125-050901

defining one or more three-dimensional collision geometry surfaces within the 3D virtual environment for defining allowed movements within the game.

11. A method as claimed in claim 8, wherein said game operation-related data includes navigation data defining a plurality of linked positions, and wherein a result of said combining includes data defining links between positions in parts of the 3D virtual environment derived from different components.

12. A method as claimed in claim 8, wherein each component has at least one viewing portal associated with an interface portion of the component for determining portions of the 3D virtual environment to render for viewing during game play, wherein said game operation-related data includes portal data defining a said viewing portal, and wherein a result of said combining comprises a combination of said portal data such that a single portal is associated with a part of the 3D virtual environment deriving from joining the interface portions of two of said components.

13. A method as claimed in claim 1, wherein the component data comprises data for at least two sets of 3D components, the method further comprising:

reading component set data identifying a said set of 3D components for use in generating said 3D virtual environment data.

14. A method of generating data for a three-dimensional (3D) virtual environment using a map of 3D components, the method comprising:

reading map data from a data store, the map data comprising component set data identifying a said set of 3D components for use in generating said 3D virtual environment data, component identifying data and component position data for said 3D components;

reading from a data store component data for the identified components from the identified set, the component data including at least 3D geometry data for the components;

transforming the 3D geometry data of the said components using said component position data; and

combining the transformed data to provide 3D virtual environment data for said 3D virtual environment.

15. A method of generating data for a three-dimensional (3D) virtual environment using a map of 3D components, each component having one or more interface portions for interfacing to others of the components, a said interface portion having an associated plug comprising plug data for providing component geometry to close the interface when the interface is not used to connect to another component; the method comprising:

reading map data from a data store, the map data comprising component identifying data and component position data for said 3D components;

reading component data for the identified components from a data store, the component data including 3D geometry data for the components;

reading plug data for a component on the map with one or more interfaces not connected to other components;

transforming the 3D geometry data of the said components using said component position data;

transforming the plug data of said component using said component position data; and

combining the transformed data to provide 3D virtual environment data for said 3D virtual environment.

16. Data processing apparatus for generating data for a 3D virtual environment using a map of 3D components, the apparatus comprising:

a data memory storing component data and operable to store map data, the map data comprising component identifying data and component position data for at least one of said 3D components, the component data including at least 3D geometry data for the at least one component;

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to:

read said map data from the data memory;

read said component data from the data memory; and

transform the 3D geometry data of the at least one said component using said component position data to provide 3D virtual environment data for said 3D virtual environment, said 3D virtual environment data defining a substantially contiguous 3D surface enclosing said 3D virtual environment.

17. A data processing apparatus as claimed in claim 16, wherein said map data comprises component identifying data and component position data for a plurality of said components, wherein said component data includes at least 3D geometry data for said plurality of components and wherein the stored instructions further comprise instructions for controlling the processor to read and transform said 3D geometry data for said plurality of components, and to combine said transformed data for said plurality of components to provide said 3D virtual environment data.

18. A data processing apparatus as claimed in claim 17, wherein each of said components has at least one interface portion for interfacing to the interface portion of another component, and wherein said component data includes two versions of said 3D geometry data for at least the interface portion of the component, a non-interfaced version of the geometry data and an interfaced version of the geometry data, and wherein the stored instructions further comprise instructions for controlling the processor to read the non-interfaced version of the geometry data for the interface portion of a component where the component is joined to another component at the interface portion.

19. A data processing apparatus as claimed in claim 16, wherein said component data for a component identified in the map data includes plug data for at least one plug for closing an interface portion of the component; and wherein the stored instructions further comprise instructions for controlling the processor to transform the plug data using component position data for the component, to close the interface when said interface of the component is not used.

20. A data processing apparatus as claimed in claim 17, wherein the three-dimensional virtual environment comprises an environment for a computer game

character, wherein the 3D geometry data comprises data for generating visible 3D geometry of the 3D virtual environment, and wherein the component data includes additional game operation-related data for defining an operational aspect of the game; and wherein the stored instructions further comprise instructions for controlling the processor to combine the game operation-related data for components identified in the map data to operationally link parts of the 3D virtual environment derived from different components.

21. Data processing apparatus for generating data for a three-dimensional (3D) virtual environment using a map of 3D components, the apparatus comprising:

a data memory operable to store map data, the map data comprising component set data identifying a said set of 3D components for use in generating said 3D virtual environment data, component identifying data and component position data for said 3D components; and storing component data including at least 3D geometry data for the components;

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to:

read said map data from said data memory;

read component data for the identified components from the identified set from said data memory;

transform the 3D geometry data of the said components using said component position data; and

combine the transformed data to provide 3D virtual environment data for said 3D virtual environment.

22. Data processing apparatus for generating data for a three-dimensional (3D) virtual environment using a map of 3D components, each component having one or more interface portions for interfacing to others of the components, a said interface portion having an associated plug comprising plug data for providing component

geometry to close the interface when the interface is not used to connect to another component; the apparatus comprising:

a data memory operable to store map data, the map data comprising component identifying data and component position data for said 3D components; and storing component data including 3D geometry data for the components and said plug data for the components;

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to:

read said map data from said data memory;

read component data for the identified components from said data memory;

read from said data memory plug data for a component on the map with one or more interfaces not connected to other components;

transform the 3D geometry data of the said components using said component position data;

transform the plug data of said component using said component position data;

and

combine the transformed data to provide 3D virtual environment data for said 3D virtual environment.

23. A method of generating a 3D virtual environment for a computer game character by assembling a plurality of 3D building blocks, each 3D building block having data defining a portion of a 3D virtual environment and an interface, the method comprising:

inputting position data defining a set of relative positions for the blocks; and

joining interfaces of the blocks using said position data, to generate a 3D virtual environment defined by a plurality of said blocks.

24. A method as claimed in claim 23, wherein said 3D virtual environment is a closed 3D virtual environment, the method further comprising:

providing data for two versions of at least the interface portion of each building block, a first version in which the interface is closed and a second version in which the interface is open; and

selecting from the first and second version of the block interface data according to whether the block is to be joined to a neighbouring block.

25. A method as claimed in claim 24, wherein the data for each building block includes visible geometry data for visible rendering of an internal space defined by the building block, wherein the first and second versions of the block interface data comprise first and second versions of said visible geometry data and wherein said second versions of a plurality of the building block interface portions defined by the block interface data of a plurality of said blocks match to provide a substantially contiguous visible internal geometry where blocks are joined.

26. A method as claimed in claim 25, wherein the data for each block further comprises viewing portal data for use in determining portions of the 3D environment defined by the blocks which can be neglected when processing portions of the 3D virtual environment for visible rendering.

27. A method as claimed in claim 24, wherein the data for each building block further includes collision geometry data defining geometry for use in determining collisions of the computer game character with features of the 3D virtual environment, and wherein the data for said first and second versions of at least the interface portion of each block defines first and second versions of said collision geometry data.

28. A method as claimed in claim 27, wherein the data for each building block further includes visible geometry data and wherein, for the first, closed version of said interface portion of a block, the collision geometry is at least partially defined using said visible geometry data.

29. A method as claimed in claim 23, wherein the data for each building block includes data defining a plurality of predetermined navigation positions for use in

navigating through the 3D environment, and wherein said joining comprises linking navigation positions of joined blocks to facilitate navigation between the blocks.

30. A method as claimed in claim 23, wherein the data for each building block includes 3D internal geometry data defining 3D internal geometry for each block, the method further comprising:

retrieving said 3D internal geometry for each block; and

using the 3D internal geometry data for the blocks to generate the 3D virtual environment.

31. A method as claimed in claim 30, wherein said blocks comprise a plurality of visually different sets of types of building blocks, each said type of 3D building block having a plurality of visually different versions, one for each set, each visually different version having data for substantially the same interface geometry but different internal visual appearance or geometry, and wherein said joining joins building blocks of a specified set, the method further comprising:

inputting data identifying said specified set of building blocks.

32. A method as claimed in claim 31, wherein each visually different version of at least one of the types of building block has data defining at least one item position for use in placing an item within the 3D virtual environment, the item position having associated item position identifier data, the item positions in the different versions of the block having substantially the same item position identifier whereby an item for the identified item position can be positioned in the block whichever version of the type of building block is selected.

33. A method as claimed in claim 23, further comprising selecting, for generating said 3D virtual environment, a set of building block data, comprising data for said plurality of building blocks, from a plurality of sets of data for said building blocks.

34. A method as claimed in claim 33, wherein the data for each of said plurality of building blocks has a version for each of said sets of building blocks, and wherein the one or more interfaces of each version of a said building block are defined by the data

as being located in substantially similar relative positions in each version of the said building block.

35. Data processing apparatus for generating a 3D virtual environment for a computer game character by assembling a plurality of 3D building blocks, each 3D building block having data defining a portion of a 3D virtual environment and an interface, the apparatus comprising:

- a data memory operable to store position data;
 - an instruction memory storing processor implementable instructions; and
 - a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;
- wherein the stored instructions comprise instructions for controlling the processor to:
- input position data defining a set of relative positions for the blocks; and
 - join interfaces of the blocks using said position data, to generate a 3D virtual environment defined by a plurality of said blocks.

36. A data processing apparatus as claimed in claim 35, wherein said 3D virtual environment is a closed 3D virtual environment and wherein the stored instructions further comprise instructions for controlling the processor to:

- provide data for two versions of at least the interface portion of each building block, a first version in which the interface is closed and a second version in which the interface is open; and
- select from the first and second version of the block interface data according to whether the block is to be joined to a neighbouring block.

37. A data processing apparatus as claimed in claim 35, wherein each visually different version of at least one of the types of building block has data defining at least one item position for use in placing an item within the 3D virtual environment, the item position having associated item position identifier data, the item positions in the different versions of the block having substantially the same item position identifier whereby an item for the identified item position can be positioned in the block whichever version of the type of building block is selected.

38. A method of constructing a three-dimensional world for a computer game using a set of predetermined three-dimensional tiles and a map specifying relative positions for the tiles, each tile comprising tile data including tile visual geometry data for defining visual geometry of the 3D world and invisible game control data; map data comprising tile identification data and tile position data identifying tiles from the set and their positions; world data comprising world visual geometry data and world game control data for the computer game, in a three-dimensional world space; the method comprising:

reading the map data;

transforming the visual geometry data into said world space, using the map data, to generate said world visual geometry data; and

transforming the invisible game control data into the world space, using the map data, to generate said world game control data.

39. A method as claimed in claim 38, wherein the invisible game control data comprises data selected from a group including collision geometry data, non-player character navigational data and viewing portal data.

40. A method as claimed in claim 39 when the game control data includes navigational data, further comprising linking navigational data from different tiles after transforming the navigational data into said world space to facilitate non-player character navigation in the world space.

41. A method as claimed in claim 39 when the game control data includes portal data, further comprising matching portal data from different tiles, after transforming the portal data into said world space, to remove duplicate portals.

42. A method as claimed in claim 38, wherein each tile has at least one interface portion for joining the tile to another tile and wherein said tile visual geometry data defines matching visual geometry for the interface portions of at least a subset of the tiles such that when tile visual geometry data for tiles of the subset is transformed into

050901 050901 050901

said world space the resulting world visual geometry data defines substantially contiguous visual world geometry across interfaces between the tiles.

43. A method as claimed in claim 38, wherein the tile data includes plug visual geometry data whereby the tile data provides data defining at least two versions of visual geometry for each tile, a first version in which an interface to the tile is closed by a visual plug defined by the plug visual geometry data and a second version in which an interface to the tile is open for joining the tile to another tile.

44. A method as claimed in claim 38, the method further comprising:
providing a user interface for constructing said map, the user interface representing the three-dimensional world as a series of two dimensional levels on which the three-dimensional tiles may be placed; and
receiving data for generating said map data from said user interface.

45. A method as claimed in claim 44, wherein said tile data includes data for one or more tiles spanning at least two of the two dimensional levels, whereby two of the two dimensional levels are linkable.

46. A method as claimed in claim 44, wherein said user interface includes an indicator for providing the user with an indication of the memory requirements of the world data for a three-dimensional world constructed using the map.

47. A method as claimed in claim 38, the method further comprising:
inputting data for selecting tile data for the said set of predetermined 3D tiles from tile data for a plurality of sets of 3D tiles, each tile within a set having tile data defining interface features for interfacing to the other tiles, the interface features of each tile substantially corresponding to interface features of at least one tile in each other set of 3D tiles.

48. Data processing apparatus for constructing a three-dimensional world for a computer game using a set of predetermined three-dimensional tiles and a map specifying relative positions for the tiles, each tile comprising tile data including the

visual geometry data for defining visual geometry of the 3D world and invisible game control data; map data comprising tile identification data and tile position data identifying tiles from the set and their positions; world data comprising world visual geometry data and world game control data for the computer game, in a three-dimensional world space; the apparatus comprising:

a data memory operable to store said map data, and storing said tile data:

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in

accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to:

read the map data;

transform the visual geometry data into said world space, using the map data, to generate said world visual geometry data; and

transform the invisible game control data into the world space, using the map data, to generate said world game control data.

49. A data processing apparatus as claimed in claim 48, wherein the tile data includes plug visual geometry data whereby the tile data provides data defining at least two versions of visual geometry for each tile, a first version in which an interface to the tile is closed by a visual plug defined by the plug visual geometry data and a second version in which an interface to the tile is open for joining the tile to another tile.

50. A data processing apparatus as claimed in claim 48, wherein the stored instructions further comprise instructions for controlling the processor to input data for selecting tile data for the said set of predetermined 3D tiles from tile data for a plurality of sets of 3D tiles, each tile within a set having tile data defining interface features for interfacing to the other tiles, the interface features of each tile substantially corresponding to interface features of at least one tile in each other set of 3D tiles.

51. A method of providing data for constructing a virtual 3D environment using predetermined 3D constructional elements, the elements having interfaces for connecting one element to another; the method comprising:

representing said 3D constructional elements to a user;
 inputting instructions from the user for assembling the elements into a structure in which the elements are connected at the interfaces, the structure representing the virtual 3D environment;
 representing the structure to the user; and
 storing structure data representing the structure on a storage medium for constructing the virtual 3D environment.

52. A method as claimed in claim 51, wherein said predetermined constructional elements comprise a plurality of sets of elements, each set having corresponding interfaces, the method further comprising:

inputting set selection data for selecting a said set of elements; and
 storing said set selection data on the storage medium in association with said structure data;

whereby data is provided specifying the construction of one of a set of said virtual 3D environments from the selected set of constructional elements.

53. A method as claimed in claim 51, further comprising:

displaying an approximate memory requirement of the virtual 3D environment represented by the structure.

54. A method as claimed in claim 53, wherein said displaying displays an approximate difference between an estimated memory requirement of the 3D virtual environment and a maximum available memory space.

55. A method as claimed in claim 51, further comprising:

inputting item placement instructions from the user; and
 storing item placement data corresponding to said item placement instructions on said storage medium in association with said structure data.

56. A method as claimed in claim 51, further comprising:

inputting lighting instructions from the user; and

storing lighting data corresponding to said lighting instructions on said storage medium in association with said structure data.

57. A method as claimed in claim 56, wherein said lighting instructions include lighting phase instructions for setting a phase of a lighting intensity variation; and wherein said lighting data includes lighting phase data corresponding to said lighting phase instructions.

58. A method as claimed in claim 51, wherein said structure data includes constructional element identification data and constructional element position data for a plurality of elements, specifying positions of said predetermined constructional elements in the structure.

59. A method as claimed in claim 58, wherein said structure further includes connection data specifying connections between constructional elements in the structure.

60. A method as claimed in claim 51, wherein said representing comprises representing said structure to the user as substantially 2D elements located on a grid, said 2D elements representing said 3D constructional elements.

61. Data processing apparatus for providing data for constructing a virtual 3D environment using predetermined 3D constructional elements, the elements having interfaces for connecting one element to another, the apparatus comprising:

a data memory operable to store structure data representing the structure for constructing the virtual 3D environment, and storing data for representing said 3D constructional elements to a user;

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to:

represent said 3D constructional elements to a user;

input instructions from the user for assembling the elements into a structure in which the elements are connected at the interfaces, the structure representing the virtual 3D environment;

represent the structure to the user; and

store said structure data on a computer readable carrier.

62. A computer readable medium carrying computer readable instructions for controlling a computer system to perform the method of any one of claims 1 to 15, 23 to 34, 38 to 47 and 51 to 66.

63. A user interface for constructing a representation of a three-dimensional virtual environment using predetermined constructional elements, each element defining a 3D geometry for use in defining a region within the 3D virtual environment, the user interface comprising:

a selection component for the user to select a said predetermined constructional element;

a placing component for the user to place the selected element in relation to other placed elements; and

a joining component to join the placed elements to provide a representation of the 3D virtual environment.

64. A user interface as claimed in claim 63, further comprising a display component to display the representation of the 3D virtual environment.

65. A user interface as claimed in claim 64, wherein the display component comprises a 2D display component to display a 2D representation of the 3D virtual environment.

66. A user interface as claimed in claim 65, wherein the 2D representation comprises a plurality of 2D grids representing sections through the 3D virtual environment, and wherein the display component includes a selection component for the user to select a said section.

09859126-050901
T06050-9273860

67. A user interface as claimed in claim 66, wherein said sections represent substantially horizontal levels within the 3D virtual environment and wherein at least one said predetermined constructional element spans two or more levels, whereby the horizontal levels are linkable.

68. A user interface as claimed in claim 64, wherein the display component comprises a 3D display component to display a 3D representation of the 3D virtual environment.

69. A user interface as claimed in claim 63, wherein each element defines a closed 3D space and wherein the joining component opens interfaces between joined elements.

70. A user interface as claimed in claim 63, wherein associated with each said constructional element is a plurality of 3D tiles, each tile belonging to a respective tileset, the tiles within each tileset having a common visual stylistic appearance, the user interface further comprising a tileset selection component for the user to select a tileset for use with said constructional elements for constructing a 3D representation of said 3D virtual environment.

71. A user interface as claimed in claim 63, further comprising a memory indicator component to provide an indication of a computer memory requirement of the 3D virtual environment.

72. A user interface as claimed in claim 63, wherein the placed and joined elements comprise a map of the 3D virtual environment, and wherein the user interface further comprises a map save component to save map data representing the map on a storage device, the map data comprising element identification data and position data.

73. Data processing apparatus for providing a user interface for constructing a representation of a three-dimensional virtual environment using predetermined constructional elements, each element defining a 3D geometry for use in defining a region within the 3D virtual environment, the apparatus comprising:

09852126-050901

a data memory storing data representing said predetermined constructional elements;

an instruction memory storing processor implementable instructions; and

a processor operable to read and process data from the data memory in accordance with instructions stored in the instruction memory;

wherein the stored instructions comprise instructions for controlling the processor to provide for said user interface:

a selection component for the user to select a said predetermined constructional element;

a placing component for the user to place the selected element in relation to other placed elements; and

a joining component to join the placed elements to provide a representation of the 3D virtual environment.

74. A data processing apparatus as claimed in claim 73, and wherein the stored instructions further comprise instructions for controlling the processor to provide for said user interface:

a display component to display the representation of the 3D virtual environment, the display component comprising a 3D display component to display a 3D representation of the 3D virtual environment.

75. A data processing apparatus as claimed in claim 73, and wherein the stored instructions further comprise instructions for controlling the processor to provide for said user interface:

a memory indicator component to provide an indication of a computer memory requirement of the 3D virtual environment.

76. A computer readable medium carrying computer readable instructions for controlling a computer system to provide the user interface of any one of claims 63 to 72.

77. A data structure on a carrier medium, the data structure comprising data for use in constructing a virtual 3D environment from predetermined constructional elements,

each constructional element having geometry defining a three-dimensional space and having at least one interface for connecting the element to another of said predetermined elements, the data structure defining an arrangement of said elements and comprising constructional element identification data and constructional element position data for each of a plurality of said elements.

78. A data structure as claimed in claim 77, further comprising connection data for each element, for determining whether the at least one interface connects to another element.

79. A data structure as claimed in claim 77, wherein said predetermined constructional elements comprise a plurality of sets of elements, each set having corresponding interfaces, the data structure further comprising set data specifying a said set of elements for use in constructing the virtual 3D environment.

80. A data structure as claimed in claim 77, further comprising object placement data for use in determining the placement of objects within the virtual 3D environment.

81. A data structure as claimed in claim 77, further comprising lighting placement data for use in determining the placement of lighting within the virtual 3D environment.

82. A data structure as claimed in claim 81, wherein said lighting data includes lighting phase data for setting a phase of a light intensity variation in the virtual 3D environment.

83. A data structure as claimed in claim 77, wherein said virtual 3D environment is a virtual environment for a computer game.

84. A data structure on a carrier medium, the data structure comprising data for a 3D constructional element for use in constructing a virtual 3D environment, the data comprising 3D geometry data defining a 3D surface for use in defining a portion of a bounding surface of said virtual 3D environment, said 3D virtual environment being located within said bounding surface.

85. A data structure as claimed in claim 84, wherein said constructional element has at least one interface portion for interfacing to another said constructional element for constructing said virtual 3D environment, wherein the 3D surface defined by said geometry data has an edge at said interface portion, said edge defining an opening, and wherein said 3D geometry data further includes plug geometry data defining a plug to close said opening, for use when said interface portion does not interface to another said constructional element.

86. A data structure as claimed in claim 84, wherein said 3D geometry data defines a 3D surface visible in said virtual 3D environment.

87. A data structure as claimed in claim 84, comprising data for a plurality of said 3D constructional elements forming a set of said constructional elements, each element within the set having at least one interface portion for interfacing to another element within the set for constructing said virtual 3D environment, said geometry data for an element defining the geometry of said interface portion of the element, the geometry of at least one interface portion of each constructional element in the set matching the geometry of an interface portion of each of the other constructional elements in the set.

88. A plurality of data structures, each as claimed in claim 87, comprising data for a plurality of said sets of constructional elements, each element having a connectivity determining to which other elements the element may be interfaced, the connectivity of an element being determined by said geometry data, each constructional element in a said set having a counterpart in each of the other sets, the connectivity of each said element substantially corresponding to the connectivity of the counterparts of the element.

89. A plurality of data structures as claimed in claim 88, wherein the geometry data for the interface portions of the constructional elements of one set is configured to define geometry to substantially match geometry of the interface portions of the elements of another said set.

09852126-050901

90. A plurality of data structures as claimed in claim 88, further comprising texture data for the constructional elements for providing a visual appearance for said 3D surface of the element, each constructional element in a said set having texture data for a themed visual appearance different to the themed visual appearance of another said set of elements.

91. A plurality of data structures as claimed in claim 88, wherein the 3D geometry defined by 3D geometry data of a constructional element in one said set differs from the 3D geometry of the counterpart to the element in another said set.

92. A data structure as claimed in claim 84, wherein the virtual 3D environment comprises an environment for a computer game and wherein said bounding surface encloses a space within which the game is played.

93. A data structure as claimed in claim 92, wherein the data for the 3D constructional element further comprises portal data defining one or more portals for use by computer game code in determining which parts of the virtual 3D environment to process for potential display.

94. A data structure as claimed in claim 93, wherein each said constructional element has an interface portion for interfacing the element to other said constructional elements, and wherein a said portal is associated with said interface portion.

95. A data structure as claimed in claim 92, wherein the data for the 3D constructional element further comprises navigation data for use by computer game code in determining a route through a part of the virtual 3D environment constructed using the 3D constructional element.

96. A data structure as claimed in claim 92, wherein said 3D geometry data includes collision geometry data for use by computer game code in determining collisions with parts of said 3D virtual environment.

97. A data structure as claimed in claim 84, wherein said constructional element has at least one interface portion for interfacing to another said constructional element for constructing said virtual 3D environment, wherein the 3D surface defined by said geometry data has an edge at said interface portion, said edge defining an opening, and wherein said 3D geometry data further includes plug geometry data defining a plug to close said opening, for use when said interface portion does not interface to another said constructional element.

98. A data structure as claimed in claim 97, wherein said plug geometry data defines both a visual geometry and a collision geometry of said 3D constructional element.

99. A method of generating data for a three-dimensional (3D) virtual object using a set of 3D components, the method comprising:

reading from a data store component identifying data and component position data for at least one of said 3D components;

reading component data for the at least one identified component from a data store, the component data including at least 3D geometry data for the component; and

transforming the 3D geometry data of the at least one component using said component position data to provide 3D virtual object data for said 3D virtual object, said 3D virtual object data defining a substantially contiguous 3D surface enclosing said 3D virtual object.

100. A method as claimed in claim 99, wherein said object position data comprises data for a plurality of said components, the method further comprising reading and transforming 3D geometry data for said plurality of components and combining said transformed data for said plurality of components to provide said 3D virtual object data.

101. A method as claimed in claim 100, wherein each of said components has at least one interface portion for interfacing to the interface portion of another component, and wherein said component data includes two versions of said 3D geometry data for at least the interface portion of the component, a non-interfaced version of the geometry data and an interfaced version of the geometry data, and wherein the method further comprises:

reading the non-interfaced version of the geometry data for the interface portion of a component where the component is joined to another component at the interface portion.

102. A method as claimed in claim 101, wherein the set of 3D components comprises at least a first subset of components which share substantially matching interfaced versions of interface portion geometry.

103. A method as claimed in claim 99, wherein said component data includes plug data for at least one plug for closing an interface portion of the component; the method further comprising transforming the plug data using component position data for the component, to close the interface when said interface of the component is not used.

104. A method as claimed in claim 100, wherein the 3D geometry data of said plurality of components comprises data for generating visible surface portions of said contiguous 3D surface enclosing said 3D virtual object.

105. A method as claimed in claim 100, wherein the three-dimensional virtual object comprises an object for a computer game, wherein the 3D geometry data comprises data for generating visible 3D geometry of the 3D virtual object, and wherein the component data includes additional object operation-related data for defining an operational aspect of the object; the method further comprising:

combining the object operation-related data to operationally link parts of the 3D virtual object derived from different components.

106. A method of generating a 3D object for a computer game by assembling a plurality of 3D building blocks, each 3D building block having data defining a portion of the 3D object and an interface, the method comprising:

inputting position data defining a set of relative positions for the blocks; and

joining interfaces of the blocks using said position data, to generate a 3D object defined by a plurality of said blocks.

107. A method as claimed in claim 106, the method further comprising:

providing data for two versions of at least the interface portion of each building block; and

selecting from the first and second version of the block interface data according to whether the block is to be joined to a neighbouring block.

108. A method as claimed in claim 107, wherein the data for each building block includes visible geometry data for visible rendering of the building block, wherein the first and second versions of the block interface data comprise first and second versions of said visible geometry data and wherein said second versions of a plurality of the building block interface portions defined by the block interface data of a plurality of said blocks match to provide a substantially contiguous visible internal geometry where blocks are joined.

109. A method as claimed in claim 106, wherein the data for each building block includes 3D geometry data defining 3D geometry for each block, the method further comprising:

retrieving said 3D geometry for each block; and

using the 3D geometry data for the blocks to generate the 3D object.

110. A method as claimed in claim 109, wherein said blocks comprise a plurality of visually different sets of types of building blocks, each said type of 3D building block having a plurality of visually different versions, one for each set, each visually different version having data for substantially the same interface geometry but different visual appearance or geometry, and wherein said joining joins building blocks of a specified set, the method further comprising:

inputting data identifying said specified set of building blocks.

111. A method as claimed in claim 106, further comprising selecting, for generating said 3D object, a set of building block data, comprising data for said plurality of building blocks, from a plurality of sets of data for said building blocks.

112. A method as claimed in claim 111, wherein the data for each of said plurality of building blocks has a version for each of said sets of building blocks, and wherein the

one or more interfaces of each version of a said building block are defined by the data as being located in substantially similar relative positions in each version of the said building block.

113. A computer readable medium carrying computer readable instructions for controlling a computer system to perform the method of any one of claims 99 to 112.

09852126-050901
T06050" 9273850